Received: 6 November 2020 | Accepted: 9 September 2021 | Published on: 21 July 2022

DOI: 10.1002/rsa.21106

RESEARCH ARTICLE

WILEY

Asymmetric Ramsey properties of random graphs involving cliques and cycles

Anita Liebenau[1] | Letícia Mattos[2] | Walner Mendonça[3] | Jozef Skokan[4]

[1]School of Mathematics and Statistics, UNSW Sydney, Sydney, New South Wales, Australia
[2]Freie Universität Berlin and Berlin Mathematical School (BMS/MATH+), Berlin, Germany
[3]IST Austria, Klosterneuburg, Austria
[4]Department of Mathematics, LSE, London, UK

Correspondence
Letícia Mattos, Freie Universität Berlin and Berlin Mathematical School (BMS/MATH+), Arnimallee 3, 14195 Berlin, Germany.
Email: leticiadmat@gmail.com

Funding information
ARC DECRA, Grant/Award Number: DE170100789; CAPES, Grant/Award Number: 88882.332408/2010-01; Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Grant/Award Numbers: EXC-2046/1, 390685689.

[Corrections added after online publication, April 17, 2023: The second sentence of the abstract has been updated.]

Abstract

We say that $G \to (F,H)$ if, in every edge coloring $c : E(G) \to \{1,2\}$, we can find either a 1-colored copy of F or a 2-colored copy of H. The well-known Kohayakawa-Kreuter conjecture states that the threshold for the property $G(n,p) \to (F,H)$ is equal to $n^{-1/m_2(F,H)}$, where $m_2(F,H)$ is given by

$$m_2(F,H) := \max\left\{\frac{e(J)}{v(J)-2+1/m_2(H)} : J \subseteq F, e(J) \geq 1\right\},$$

for any pair of graphs F and H with $m_2(F) \geq m_2(H)$. In this article, we show the 0-statement of the Kohayakawa–Kreuter conjecture for every pair of cycles and cliques.

KEYWORDS

Kohayakawa-Kreuter conjecture, Ramsey theory, random graphs

1 | INTRODUCTION

We say that a graph G is a *Ramsey graph for the pair of graphs* (F,H) if, in every edge coloring $c : E(G) \to \{1,2\}$, we can find either a 1-colored copy of F or a 2-colored copy of H. We write $G \to (F,H)$ if G is Ramsey for (F,H), and $G \nrightarrow (F,H)$ otherwise. It follows from Ramsey's theorem [11] that, for each pair of graphs (F,H), there exists a graph G such that $G \to (F,H)$.

The study of whether or not the binomial random graph $G(n,p)$ is Ramsey for a symmetric pair of graphs was initiated by Frankl and Rödl [2], and Łuczak et al. [8]. They showed that the probability threshold for having $G(n,p) \to (K_3, K_3)$ is of order $n^{-1/2}$. In 1995, Rödl and Ruciński [12, 13] determined the probability threshold for $G(n,p) \to (F,F)$ for almost all non-empty graphs F. They showed that, if F has a component which is not a a star or a path of length three, then the threshold is of order $n^{-1/m_2(F)}$, where

$$m_2(F) := \max\left\{\frac{e(J)-1}{v(J)-2} : J \subseteq F, v(J) \geq 3\right\}.$$

The parameter $m_2(F)$ is called the m_2-density of the graph F. Here, $v(J)$ and $e(J)$ denote the size of the vertex set and of the edge set of the graph J, respectively. The remaining cases were addressed subsequently by Friedgut and Krivelevich [3].

A natural generalization of this problem is to determine a threshold function $p(F,H)$ for the property $G(n,p) \to (F,H)$, for any asymmetric pair of graphs (F,H). This problem was posed in 1997 by Kohayakawa and Kreuter [5], who proved that $p(C_\ell, C_k) = \Theta(n^{1-\ell((\ell-1)k)^{-1}})$ for any pair of cycles (C_ℓ, C_k) with $k \geq \ell \geq 3$. In the same paper, they conjectured that $p(F,H) = \Theta(n^{-1/m_2(F,H)})$, where

$$m_2(F,H) := \max\left\{\frac{e(J)}{v(J)-2+1/m_2(H)} : J \subseteq F, e(J) \geq 1\right\},$$

for any pair of graphs F and H with $m_2(F) \geq m_2(H) \geq 1$. Since the Kohayakawa–Kreuter conjecture was posed, there have been many attempts to solve it (see, e.g., [4, 6, 9]). In a recent breakthrough, Mousset et al. [10] showed that $p(F,H) = O(n^{-1/m_2(F,H)})$ whenever $m_2(F) \geq m_2(H) \geq 1$, the so-called *1-statement*. In contrast, much less is known about the *0-statement*, that is, the statement that $p(F,H) = \Omega(n^{-1/m_2(F,H)})$ whenever $m_2(F) \geq m_2(H) \geq 1$. One possible reason for that is that the 0-statement seems to depend on the structural behavior of Ramsey graphs.

Previously, the 0-statement was only proved for two types of pairs of graphs. Kohayakawa and Kreuter [5] established the 0-statement for all pairs of cycles while Marciniszyn et al. [9] addressed all pairs of cliques.

In this article, we show that the 0-statement holds for any pair of cliques and cycles. This is the first 0-statement result for different types of graphs.

Theorem 1.1. *For all $\ell, r \geq 4$ there exists $c > 0$ such that, if $p = p(n) \leq cn^{-1/m_2(K_r, C_\ell)}$, then*

$$\lim_{n\to\infty} \mathbb{P}\left[G(n,p) \to (K_r, C_\ell)\right] = 0.$$

Combining Theorem 1.1 with the results of [5, 9, 10], we establish the Kohayakawa–Kreuter conjecture for any pair of cycles and cliques with at least 3 vertices. We remark that we do need the assumption $\ell, r \geq 4$ in our proof, so our result does not imply the earlier results involving K_3.

The main tool behind the proof of Theorem 1.1 is a structural characterization of Ramsey graphs for the pair (K_r, C_ℓ) via a 'container type' argument (see Theorem 2.1), which is a rephrasing of an important idea used in previous works. Roughly speaking, we find a family $\mathcal{I}$ of graphs with the following properties: (a) $|\mathcal{I}|$ is small; (b) for every graph G with $G \to (K_r, C_\ell)$ there exists $I \in \mathcal{I}$ such that $I \subseteq G$; and (c) for each $I \in \mathcal{I}$, either I is small and dense or very structured. We provide the details in Section 2.

The rest of the article is organized as follows. In Section 2, we prove Theorem 1.1; in Section 3, we provide the main technical lemmas of this article; in Section 4, we prove some structural lemmas about Ramsey graphs; in Section 5, we describe the algorithms used to prove our main technical theorem (see Theorem 2.1); finally, in Section 6, we do a careful analysis of these algorithms. In the Appendix, we provide some simple calculations involving m_2-densities, for completeness.

2 | THE MAIN TECHNICAL RESULT

In this section, we present the main technical result of this article and deduce Theorem 1.1 from it. In order to state this result, we need the following notation. For a graph G, define $\lambda(G)$ by

$$\lambda(G) = v(G) - \frac{e(G)}{m_2(K_r, C_\ell)}.$$

This parameter plays an important role in our analysis, as the expected number of copies of G in $G(n,p)$ is of order $c^{e(G)}n^{\lambda(G)}$ when $p = cn^{-1/m_2(K_r,C_\ell)}$. For any positive real numbers M, ε and any positive integer n, define

$$\mathcal{J}_1(\varepsilon) = \{G : \lambda(G) \leq -\varepsilon\} \qquad \text{and} \qquad \mathcal{J}_2(M,n) = \{G : \lambda(G) \leq M \ \text{ and } \ e(G) \geq \log n\}, \tag{1}$$

where the logarithm is in base 2. Finally, for any natural numbers r, ℓ, and n, let

$$\mathcal{R}_n(K_r, C_\ell) = \{G : V(G) = [n] \text{ and } G \to (K_r, C_\ell)\},$$

where $[n] := \{1, 2, \dots, n\}$. When r and ℓ are clear from context, we write $\mathcal{R}_n$ for $\mathcal{R}_n(K_r, C_\ell)$. In addition, we set

$$\mathcal{R}(K_r, C_\ell) = \bigcup_{n \in \mathbb{N}} \mathcal{R}_n(K_r, C_\ell).$$

The connection between λ, $\mathcal{J}_1(\varepsilon)$, $\mathcal{J}_2(M,n)$, and $\mathcal{R}_n$ is contextualized in the next theorem.

Theorem 2.1. *For any integers $r, \ell \geq 4$, there exist positive constants $M = M(r,\ell)$ and $\varepsilon = \varepsilon(r,\ell)$ such that the following holds. For every positive integer n, there exists a function $f : \mathcal{R}_n(K_r, C_\ell) \to \mathcal{J}_1(\varepsilon) \cup \mathcal{J}_2(M,n)$ such that G contains a copy of $f(G)$ as a subgraph, for all $G \in \mathcal{R}_n$, and*

$$|f(\mathcal{R}_n)| \leq (\log n)^M.$$

In the language of hypergraph containers [1, 14], Theorem 2.1 provides a relatively small collection $f(\mathcal{R}_n)$ of *fingerprints*. Additionally to $|f(\mathcal{R}_n)|$ being small, each graph $f(G)$ either has a *very* small value of λ (negative, and bounded away from 0), or a fairly small (though possibly positive) value of λ and is *very* large. To obtain such a collection and the function f in Theorem 2.1, we employ an algorithm adapted from [7].

Theorem 1.1 is easily deduced from Theorem 2.1. The proof of Theorem 2.1 is given in the next four sections.

Proof of Theorem 1.1. Given $r, \ell \geq 4$, let M and ε be positive constants given by Theorem 2.1 and set $c = 2^{-2M}$. For each $n \in \mathbb{N}$, let $p = p(n) \leq cn^{-1/m_2(K_r,C_\ell)}$. Let f be the function given by Theorem 2.1, let $\Gamma \sim G(n,p)$ and suppose that $\Gamma \in \mathcal{R}_n$. Then, $f(\Gamma) \subseteq \Gamma$ and $f(\Gamma) \in \mathcal{J}_1(\varepsilon) \cup \mathcal{J}_2(M,n)$. Let $\mathcal{I}_1 := f(\mathcal{R}_n) \cap \mathcal{J}_1(\varepsilon)$ and $\mathcal{I}_2 := f(\mathcal{R}_n) \cap \mathcal{J}_2(M,n)$. Thus,

$$\mathbb{P}(\Gamma \to (K_r, C_\ell)) \leq \mathbb{P}(F \subseteq \Gamma \ \text{ for some } \ F \in \mathcal{I}_1 \cup \mathcal{I}_2). \tag{2}$$

Since $\lambda(F) \leq -\varepsilon$ for each $F \in \mathcal{I}_1$ and $c \leq 1$, we have

$$\mathbb{P}(F \subseteq \Gamma) \leq n^{v(F)}p^{e(F)} \leq c^{e(F)}n^{\lambda(F)} \leq n^{-\varepsilon}, \tag{3}$$

for every $F \in \mathcal{I}_1$. Similarly, we have

$$\mathbb{P}(F \subseteq \Gamma) \leq c^{e(F)} n^{\lambda(F)} \leq n^{-M}, \tag{4}$$

for every $F \in \mathcal{I}_2$, as $\lambda(F) \leq M$ and $e(F) \geq \log n$ for each $F \in \mathcal{I}_2$, and by our choice of c. Applying the union bound to (2) and using (3) and (4), we obtain that

$$\mathbb{P}\,(G(n,p) \to (K_r, C_\ell)) \leq (\log n)^M \cdot (n^{-\varepsilon} + n^{-M}),$$

since $|\mathcal{I}_1 \cup \mathcal{I}_2| \leq (\log n)^M$. As the expression on the right hand side tends to 0 as $n \to \infty$, this implies the theorem. ∎

3 | PROOF OF THEOREM 2.1

In this section, we state the main technical lemmas of this article, and deduce Theorem 2.1 from them. We also introduce some notation that we use during the proof.

Let r, ℓ be positive integers. We follow the approach by [5, 7] and bring our problem into the hypergraph setting. Given a graph $G = (V, E)$, let $\mathcal{G}_{r,\ell}(G)$ be the hypergraph on the edge set of G whose hyperedges correspond to the copies of K_r and C_ℓ in G. We suppress G, r and ℓ from the notation whenever they are clear from context. Define

$$\mathcal{E}_1(\mathcal{G}) = \{E(F) : F \cong K_r, F \subseteq G\} \text{ and } \mathcal{E}_2(\mathcal{G}) = \{E(F) : F \cong C_\ell, F \subseteq G\}. \tag{5}$$

Analogously, if $\mathcal{H} \subseteq \mathcal{G}_{r,\ell}(G)$, then we set $\mathcal{E}_1(\mathcal{H}) := \mathcal{E}_1(\mathcal{G}) \cap E(\mathcal{H})$ and $\mathcal{E}_2(\mathcal{H}) := \mathcal{E}_2(\mathcal{G}) \cap E(\mathcal{H})$.

The reason why we deal with $\mathcal{G}_{r,\ell}(G)$ instead of G is as follows. In order to build our fingerprints algorithmically, we would like to deal with a subgraph H of G which has the two following properties: (1) every edge $e \in E(H)$ is contained in an ℓ-cycle; (2) for *every* copy C of C_ℓ in H end every $e \in C$, there exists a copy K of K_r such that $E(K) \cap E(C) = \{e\}$. These properties would allow us to build the fingerprint of G algorithmically by attaching either a copy of K_r or a copy of C_ℓ to the current graph at each step. However, such a graph H might not exist. Even if H is minimal (with respect to subgraph containment) for $H \to (K_r, C_\ell)$, we can only deduce that for every $e \in E(H)$ there exist $K \cong K_r$ and $C \cong C_\ell$ in H such that $E(K) \cap E(C) = \{e\}$. But this does not directly imply that property (2) holds. We can overcome this problem by considering subhypergraphs of $\mathcal{G}_{r,\ell}(G)$ which are $\star$-critical. This property was first considered in [7].

Definition 3.1 ($\star$-critical)**.** Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two families of sets on a vertex set. We say that a hypergraph $\mathcal{H} = \mathcal{E}_1 \cup \mathcal{E}_2$ is $\star$-critical with respect to $(\mathcal{E}_1, \mathcal{E}_2)$ if the following two properties hold. For each $e \in V(\mathcal{H})$, there exists a hyperedge $F \in \mathcal{E}_2$ such that $e \in F$; and for each $F \in \mathcal{E}_2$ and each $e \in F$, there exists a hyperedge $E \in \mathcal{E}_1$ such that $E \cap F = \{e\}$. When $\mathcal{E}_1$ and $\mathcal{E}_2$ are clear from context, we say that the hypergraph $\mathcal{H}$ is $\star$-critical.

Let $\mathrm{Crit}_{r,\ell}(G)$ be the set of all $\star$-critical subhypergraphs of $\mathcal{G}_{r,\ell}(G)$. The next lemma shows that if $G \to (K_r, C_\ell)$, then there are subhypergraphs of $\mathcal{G}_{r,\ell}(G)$ which are $\star$-critical. We prove it in Section 4.

Lemma 3.2. *Let $r, \ell \geq 4$ be integers. If $G \to (K_r, C_\ell)$, then* $\mathrm{Crit}_{r,\ell}(G) \neq \emptyset$.

Given a hypergraph $\mathcal{H} \subseteq \mathcal{G}_{r,\ell}(G)$, we define the *underlying graph of* $\mathcal{H}$, denoted by $\mathbf{G}(\mathcal{H})$, to be the subgraph of G whose edge set is $\bigcup_{E \in E(\mathcal{H})} E$. The following lemma is central to our proof. We prove it in Section 4.

Lemma 3.3. *Let $r, \ell \geq 4$ be integers. There exists $\varepsilon = \varepsilon(r, \ell) > 0$ such that the following holds for any graph H. If $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(H)$, then $\lambda(\mathbf{G}(\mathcal{H})) \leq -\varepsilon$.*

In order to find the function $f : \mathcal{R}_n(K_r, C_\ell) \to \mathcal{J}_1(\varepsilon) \cup \mathcal{J}_2(M, n)$ in Theorem 2.1, we define an algorithm Hypertree in Section 5. For each $G \in \mathcal{R}_n(K_r, C_\ell)$, this algorithm takes some hypergraph $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$ as input and creates a subhypergraph $\mathcal{H}_T \subseteq \mathcal{H}$ as output. The fingerprint $f(G)$ will be a graph isomorphic to $\mathbf{G}(\mathcal{H}_T)$. For the detailed description of Hypertree, we refer the reader to Section 5.

Let Hypertree($\mathcal{H}$) denote the execution of Hypertree on input $\mathcal{H}$. Its basic properties are given by the next lemma.

Lemma 3.4. *Let $n, r, \ell \geq 4$ be integers and G be a graph on vertex set $[n]$. For any hypergraph $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$, Hypertree($\mathcal{H}$) generates a sequence of hypergraphs $\mathcal{H}_0 \subseteq \cdots \subseteq \mathcal{H}_T \subseteq \mathcal{H}$ for which the following holds.*

(a) $\mathcal{H}_0 = \{E\}$, for some $E \in \mathcal{E}_1(\mathcal{H})$; that is, the underlying graph of $\mathcal{H}_0$ is a copy of K_r in G;
(b) $v(\mathcal{H}_0) < v(\mathcal{H}_1) < \cdots < v(\mathcal{H}_T)$;
(c) T is the smallest integer such that $\lambda(G_T) \leq -\varepsilon$ or $T \geq \log n$, where $G_T = \mathbf{G}(\mathcal{H}_T)$ and ε is the constant given by Lemma 3.3;
(d) Hypertree($\mathcal{H}$) returns $\mathcal{H}_T$.

Our next lemma is one of the most important properties of the Hypertree algorithm. We shall use it together with Lemma 3.4 to deduce that the underlying graph given by the output of Hypertree($\mathcal{H}$) belongs to $\mathcal{J}_1(\varepsilon) \cup \mathcal{J}_2(M, n)$ whenever $\mathcal{H}$ is a hypergraph in $\mathrm{Crit}_{r,\ell}(G)$, where $M = M(r, \ell) > 0$. As the underlying graph of the output hypergraph is a subgraph of G, this will establish the existence of a function $f : \mathcal{R}_n(K_r, C_\ell) \to \mathcal{J}_1(\varepsilon) \cup \mathcal{J}_2(M, n)$ as required for Theorem 2.1.

Lemma 3.5. *Let $r, \ell \geq 4$ be integers, G a graph and $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$. Let $(\mathcal{H}_i)_{i=0}^T$ be the sequence generated by Hypertree($\mathcal{H}$) and let $G_i = \mathbf{G}(\mathcal{H}_i)$ for each $i \in \{0, \ldots, T\}$. Then we have*

$$\lambda(G_i) \leq \lambda(G_{i-1}),$$

for each $i \in \{1, \ldots, T\}$.

We prove a stronger version of this lemma (Lemma 6.1) in Section 6. For each $n, r, \ell \geq 4$, consider the family of possible output graphs of Hypertree($\mathcal{H}$)

$$\mathrm{Out}_{r,\ell}(n) = \bigcup_{G \,:\, V(G)=[n]} \left\{ \mathbf{G}(\mathcal{H}_T) \ : \ \mathcal{H} \in \mathrm{Crit}_{r,\ell}(G) \right\},$$

where $T = T(\mathcal{H})$ and $\mathcal{H}_T$ are the stopping time and the output given by Hypertree($\mathcal{H}$), respectively. The set $\mathrm{Out}_{r,\ell}(n)$ will, of course, be rather large and we cannot hope for $|\mathrm{Out}_{r,\ell}(n)|$ to be bounded by a function that is poly-logarithmic in n. Instead, we restrict to our attention to isomorphism classes. For a set S of graphs, denote by $S/_{\cong}$ a set consisting of exactly one representative graph for every (graph) isomorphism class of S. The next lemma bounds the size of $\mathrm{Out}_{r,\ell}(n)/_{\cong}$.

Lemma 3.6. *For all $r, \ell \geq 4$, there exists $C > 0$ such that $|\mathrm{Out}_{r,\ell}(n)/_{\cong}| \leq (\log n)^C$, for all $n \in \mathbb{N}$.*

We prove this lemma in Section 6. Now, we are ready to prove Theorem 2.1 assuming all the lemmas stated in this section.

Proof of Theorem 2.1. Fix $n, r, \ell \geq 4$. For each $G \in \mathcal{R}_n(K_r, C_\ell)$, let $\mathcal{H}(G)$ be a $\star$-critical hypergraph in $\mathrm{Crit}_{r,\ell}(G)$. By Lemma 3.2, such a hypergraph must exist. Let $(\mathcal{H}_i(G))_{i=0}^T$ be the sequence of hypergraphs generated by HYPERTREE($\mathcal{H}(G)$). By Lemma 3.4, the last hypergraph of this sequence, namely $\mathcal{H}_T(G)$, is the hypergraph output by HYPERTREE($\mathcal{H}(G)$).

Define

$$\begin{aligned} f : \mathcal{R}_n &\to \mathrm{Out}_{r,\ell}(n)/_{\cong} \\ G &\mapsto [\mathbf{G}(\mathcal{H}_T(G))], \end{aligned}$$

where by $[\mathbf{G}(\mathcal{H}_T(G))]$ we denote the graph $G' \in \mathrm{Out}_{r,\ell}(n)/_{\cong}$ isomorphic to $\mathbf{G}(\mathcal{H}_T(G))$. As $\mathcal{H}_T(G) \subseteq \mathcal{H}(G)$ by Lemma 3.4, and $\mathbf{G}(\mathcal{H}(G)) \subseteq G$ by construction, we have that G contains a copy of $f(G)$ as a subgraph for each $G \in \mathcal{R}_n$. Moreover, by Lemma 3.4(c), we have $\lambda(f(G)) \leq -\varepsilon$ or $T \geq \log n$. In the first case, $f(G)$ belongs to the set of graphs

$$\mathcal{J}_1(\varepsilon) = \{H : \lambda(H) \leq -\varepsilon\}.$$

In the second case, we claim that $f(G) \in \mathcal{J}_2(C, n)$, where $C = \lambda(K_r)$. To see that, first note that the sequence $(v(\mathcal{H}_i(G)))_{i=0}^T$ is strictly increasing by Lemma 3.4(b). For simplicity, let $G_i = \mathbf{G}(\mathcal{H}_i(G))$. Since $e(G_i) = v(\mathcal{H}_i(G))$ for every $i \in \{0, \ldots, T\}$, we have

$$e(f(G)) = v(\mathcal{H}_T(G)) \geq T \geq \log n. \tag{6}$$

Moreover, by Lemma 3.5, we have $\lambda(G_i) \leq \lambda(G_{i-1})$ for each $i \in \{0, \ldots, T\}$, where $G_0 \cong K_r$ by Lemma 3.4 (a). In particular,

$$\lambda(f(G)) = \lambda(G_T) \leq \lambda(G_0) = \lambda(K_r). \tag{7}$$

Together, (6) and (7) imply that $f(G) \in \mathcal{J}_2(C, n)$, where $C = \lambda(K_r)$. This proves our claim.

Now, it only remains to show that $|f(\mathcal{R}_n)| \leq (\log n)^{C_0}$, for some constant $C_0 > 0$. But, this follows directly from Lemma 3.6. We finish the proof by setting $M = \max\{C_0, C\}$. ■

4 | THE STRUCTURAL LEMMAS

In this section, we obtain some key structural information about Ramsey hypergraphs and prove Lemmas 3.2 and 3.3.

Given two families of sets $\mathcal{E}_1$ and $\mathcal{E}_2$ on a vertex set and a hypergraph $\mathcal{H}$, define

$$\mathcal{E}_1(\mathcal{H}) = \mathcal{E}_1 \cap E(\mathcal{H}) \qquad \text{and} \qquad \mathcal{E}_2(\mathcal{H}) = \mathcal{E}_2 \cap E(\mathcal{H}).$$

We refer to the hyperedges of $\mathcal{E}_1(\mathcal{H})$ and $\mathcal{E}_2(\mathcal{H})$ as, respectively, hyperedges of type 1 and 2. We say that $\mathcal{H}$ is Ramsey for $(\mathcal{E}_1, \mathcal{E}_2)$, and we write $\mathcal{H} \to (\mathcal{E}_1, \mathcal{E}_2)$, if the following holds. For every

2-colouring $c : V(\mathcal{H}) \to \{1, 2\}$, there exists a hyperedge $E \in \mathcal{E}_i(\mathcal{H})$ such that $c(E) = \{i\}$, for some $i \in \{1, 2\}$. Conversely, we write $\mathcal{H} \nrightarrow (\mathcal{E}_1, \mathcal{E}_2)$ if $\mathcal{H} \to (\mathcal{E}_1, \mathcal{E}_2)$ is not satisfied. Clearly, if $\mathcal{H} \subseteq \mathcal{F}$ and $\mathcal{H} \to (\mathcal{E}_1, \mathcal{E}_2)$, then $\mathcal{F} \to (\mathcal{E}_1, \mathcal{E}_2)$. Therefore, we may concentrate on the minimal hypergraphs $\mathcal{H}$ that satisfy $\mathcal{H} \to (\mathcal{E}_1, \mathcal{E}_2)$.

We call a hypergraph $\mathcal{H}$ *Ramsey minimal* with respect to $(\mathcal{E}_1, \mathcal{E}_2)$ if $\mathcal{H} \to (\mathcal{E}_1, \mathcal{E}_2)$, yet the removal of any hypervertex or hyperedge from $\mathcal{H}$ destroys this property. Minimal Ramsey hypergraphs have the $\star$-critical property, as the next lemma shows. Its proof follows the same steps as the proof of [7, Claim 1]. A particular case of it can be also found in [5, Section 3].

Lemma 4.1. *Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two disjoint families of sets on a vertex set. If a hypergraph $\mathcal{H}$ is Ramsey minimal with respect to $(\mathcal{E}_1, \mathcal{E}_2)$, then the following holds. For each $i \in \{1, 2\}$, each hyperedge $E \in \mathcal{E}_i$, and each hypervertex $e \in E$, there exists a hyperedge $F \in \mathcal{E}_{3-i}$ such that $E \cap F = \{e\}$. In particular, $\mathcal{H}$ is $\star$-critical.*

Proof. Fix any hyperedge $E \in \mathcal{E}_i(\mathcal{H})$, for some $i \in \{1, 2\}$, and any hypervertex $e \in E$. Let $\mathcal{H} \setminus E$ be the hypergraph with vertex set $V(\mathcal{H})$ and hyperedge set $E(\mathcal{H}) \setminus \{E\}$. Consider any coloring $c : V(\mathcal{H}) \to \{1, 2\}$ for which there is no hyperedge of type j in $\mathcal{H} \setminus E$ colored j under c, for all $j \in \{1, 2\}$. This coloring exists because $\mathcal{H}$ is Ramsey minimal. As $\mathcal{H} \to (\mathcal{E}_1, \mathcal{E}_2)$, all the hypervertices in E must be colored i under c. Moreover, E is the only monochromatic hyperedge under c which has type j and color j, for $j = 1, 2$. Now, let $c' : V(\mathcal{H}) \to \{1, 2\}$ be the coloring such that $c'(f) = c(f) \Leftrightarrow f \neq e$ (recall that $e \in E$). As $\mathcal{H} \to (\mathcal{E}_1, \mathcal{E}_2)$ and E is not monochromatic of color i under c', there must exist a hyperedge $F \in \mathcal{E}_{3-i}(\mathcal{H})$ such that $c'_E(F) = 3 - i$ and $E \cap F = \{e\}$, as required. ■

Now we are ready to prove Lemma 3.2.

Proof of Lemma 3.2. If $G \to (K_r, C_\ell)$, then $\mathcal{G}_{r,\ell}(G) \to (\mathcal{E}_1, \mathcal{E}_2)$, where $\mathcal{E}_1$ and $\mathcal{E}_2$ are defined in (5). Let $\mathcal{H}$ be an arbitrary Ramsey minimal subhypergraph of $\mathcal{G}_{r,\ell}(G)$. Then $\mathcal{H}$ is $\star$-critical, by Lemma 4.1, and therefore $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$, as required. ■

We now turn to the proof of Lemma 3.3. In order to prove it, we require some structural information about underlying graphs of $\star$-critical hypergraphs. This structural information is obtained in Lemma 4.3 and, before stating it, it is worth to point out the following observation.

Observation 4.2. Let $r, \ell \geq 3$. For any graph H and any hypergraph $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(H)$, we have $d(v) \geq r$ for all $v \in V(\mathbf{G}(\mathcal{H}))$.

In fact, if $\mathcal{H}$ is $\star$-critical, then for any $e \in E(\mathbf{G}(\mathcal{H}))$ there exists $K \in \mathcal{E}_1(\mathcal{H})$ and $C \in \mathcal{E}_2(\mathcal{H})$ such that $K \cap C = \{e\}$. As K and C are copies of K_r and C_ℓ contained in $\mathbf{G}(\mathcal{H})$, respectively, we can easily infer that $d(v) \geq r$ for all $v \in V(\mathbf{G}(\mathcal{H}))$.

Now, we need to set some notation. For each graph G, define

$$A = A(G) = \{v \in V(G) : d(v) = r\} \qquad \text{and} \qquad B = B(G) = \{v \in V(G) : d(v) > r\}. \tag{8}$$

By Observation 4.2, $V(G)$ can be partitioned into $V(G) = A \cup B$ whenever G is the underlying graph of a $\star$-critical hypergraph. Below, we use $N(v)$ to denote the neighborhood of a vertex v in G and, for each $S \subseteq V(G)$, we write $d_S(v) = |N(v) \cap S|$. Our structural lemma is as follows.

Lemma 4.3. *Let $r \geq 3$ and $\ell \geq 4$ be integers, and H be a graph. For any hypergraph $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(H)$, we have that*

(1) A is an independent set in $\mathbf{G}(\mathcal{H})$ and
(2) $d_B(v) \geq r-2$, for all $v \in V(\mathbf{G}(\mathcal{H}))$.

Proof. First, let us prove item (1). Suppose for a contradiction that there are two adjacent vertices $u, v \in V(\mathbf{G}(\mathcal{H}))$ such that $d(u) = d(v) = r$. As $\mathcal{H}$ is $\star$-critical, there exists an r-clique $R_1 \in \mathcal{E}_1(\mathcal{H})$ and an ℓ-cycle $C_1 \in \mathcal{E}_2(\mathcal{H})$ such that $E(R_1) \cap E(C_1) = \{uv\}$. Now, let u_c be the neighbor of u in $C_1 \setminus \{v\}$ and v_c be the neighbor of v in $C_1 \setminus \{u, v\}$. First, we observe that $u_c \notin N(v)$ and $v_c \notin N(u)$. Indeed, note that $N(v) = (V(R_1) \setminus \{v\}) \cup \{v_c\}$ and $\ell \geq 4$ implies that $u_c \neq v_c$, hence $u_c \notin N(v)$. Similarly, we have $v_c \notin N(u)$. This will be used in the rest of the proof below.

Now, fix any vertex $w \in V(R_1) \setminus \{u, v\}$. We have the following claim.

Claim 4.4. There exists an ℓ-cycle $C_2 \in \mathcal{E}_2(\mathcal{H})$ such that either $\{u_c u, uw\} \subseteq E(C_2)$ or $\{vu, uw\} \subseteq E(C_2)$.

Proof of the Claim. As $\mathcal{H}$ is $\star$-critical, there exists an r-clique R_2 and an ℓ-cycle C_2 such that $E(R_2) \cap E(C_2) = \{uw\}$. If $R_2 = R_1$, then C_2 must contain u_c, as $d(u) = r$. This settles the first part of the claim. If $R_2 \neq R_1$, then R_2 must contain u_c, again because $d(u) = r$. As $u_c \in V(R_2)$, we cannot have $v \in R_2$. Otherwise, we would have $u_c \in N(v)$, which is a contradiction. Together, these imply that $V(R_2) = \{u_c\} \cup V(R_1) \setminus \{v\}$. As v is the only vertex in $V(R_1)$ not contained in R_2, it follows that $v \in C_2$. This settles the second part. ■

Let C_2 be the cycle given by the claim above. If $\{u_c u, uw\} \subseteq E(C_2)$, then there exists an r-clique $R_2 \in \mathcal{E}_1(\mathcal{H})$ such that $E(R_2) \cap E(C_2) = \{u_c u\}$. As $V(R_2) \subseteq N(u) \cup \{u\} \setminus \{w\}$, R_2 has no choice but to contain v. In particular, this implies that v is a neighbor of u_c, which gives us a contradiction. If $\{vu, uw\} \subseteq E(C_2)$, then there exists an r-clique $R_2 \in \mathcal{E}_1(\mathcal{H})$ such that $E(R_2) \cap E(C_2) = \{uv\}$. As $V(R_2) \subseteq N(u) \cup \{u\} \setminus \{w\}$, R_2 has no choice but to contain u_c. In particular, this implies again that v is a neighbor of u_c, which gives us a contradiction. This proves item (1).

To show item (2), we consider two cases: (i) either $d_B(v) = d(v)$, or (ii) $d(v) > d_B(v)$. In the first case, Observation 4.2 gives us $d_B(v) = d(v) \geq r$. In the second case, there is a vertex $u \in N(v) \cap A$ and, since $\mathcal{H}$ is $\star$-critical, there is also an r-clique R in $\mathbf{G}(\mathcal{H})$ such that $uv \in E(R)$. As A is an independent set, we must have $V(R) \setminus \{u\} \subseteq B$, which implies that v has least $r-2$ neighbors in B. ■

Let $m(G) = e(G)/v(G)$ be the edge density of G. Now, we are ready to prove Lemma 3.3.

Proof of Lemma 3.3. To simplify the notation, set $G = \mathbf{G}(\mathcal{H})$. We show that there exists $\delta > 0$ such that $m(G) > m_2(K_r, C_\ell) + \delta$. Note that this implies that $m(G) - m_2(K_r, C_\ell) \geq \varepsilon m_2(K_r, C_\ell) v(G)^{-1}$ for $\varepsilon = \delta / m_2(K_r, C_\ell)$, as $|v(G)| \geq 1$. This can be seen to be equivalent to $\lambda(G) \leq -\varepsilon$, by definition of λ.

In order to find δ, we shall first bound $e(G)$ from below. By Observation 4.2, the set $V(G)$ can be partitioned into $V(G) = A \cup B$, where $A = A(G)$ and $B = B(G)$ were defined in (8). Thus, we can write

$$2e(G) = \sum_{v \in A} d(v) + \sum_{v \in B} d(v).$$

As $d(v) = r$ for all $v \in A$, we have $\sum_{v \in A} d(v) = r|A|$. Now, to bound the sum $S = \sum_{v \in B} d(v)$, observe that this sum counts twice each edge inside the set B and counts once each edge across A and B. By Lemma 4.3(1), we have $e(A, B) = r|A|$, as A is an independent set and $d(v) = r$ for each $v \in A$. By Lemma 4.3(2), $d_B(v) \geq r-2$ for each $v \in B$. Together, these imply that

$$S \geq r|A| + (r-2)|B|.$$

Furthermore,

$$S \geq (r+1)|B|,$$

as $d(v) \geq r+1$ for each $v \in B$. Therefore,

$$2e(G) \geq r|A| + \max\{r|A| + (r-2)|B|, (r+1)|B|\}.$$

As $v(G) = |A| + |B|$, we have

$$\begin{aligned} 2m(G) &\geq \max\left\{\frac{2r|A| + (r-2)|B|}{|A| + |B|}, \frac{r|A| + (r+1)|B|}{|A| + |B|}\right\} \\ &= r - 2 + \max\{(r+2)x, 3 - x\}, \end{aligned}$$

where $x = |A|/v(G)$. The last expression attains its minimum value when $x = 3/(r+3)$, and hence

$$m(G) \geq \frac{r+1}{2} - \frac{3}{2(r+3)}.$$

A straightforward calculation shows that $m_2(K_r, C_\ell) = \frac{\binom{r}{2}(\ell-1)}{(r-1)(\ell-1)-1}$ (see Fact 6.2). From this expression, we can see that $\ell \mapsto m_2(K_r, C_\ell)$ is decreasing. Thus, in order to conclude our proof, it suffices to show that

$$\frac{r+1}{2} - \frac{3}{2(r+3)} > m_2(K_r, C_4).$$

Using again the expression we have for $m_2(K_r, C_\ell)$, an easy calculation shows that the last inequality holds for every $r \geq 4$. This completes the proof of the lemma. ■

5 | THE ALGORITHMS

In this section, we formally describe the algorithm HYPERTREE and its subroutine FLOWER, and prove Lemma 3.4. Let $n, r, \ell \geq 4$ be fixed integers throughout this section.

First, let us recall some notation from Section 3. Given any graph G, $\mathrm{Crit}_{r,\ell}(G)$ denotes the set of all $\star$-critical subhypergraphs of $\mathcal{G}_{r,\ell}(G)$, the hypergraph whose hyperedges correspond to the copies of K_r and C_ℓ in G. We distinguish the hyperedges of $\mathcal{G}_{r,\ell}(G)$ by two types:

$$\mathcal{E}_1(\mathcal{G}_{r,\ell}(G)) = \{E(F) : F \cong K_r, F \subseteq G\} \text{ and } \mathcal{E}_2(\mathcal{G}_{r,\ell}(G)) = \{E(F) : F \cong C_\ell, F \subseteq G\}.$$

For any $\mathcal{H} \subseteq \mathcal{G}_{r,\ell}(G)$, we denote $\mathcal{E}_1(\mathcal{H}) = E(\mathcal{H}) \cap \mathcal{E}_1(\mathcal{G}_{r,\ell}(G))$ and $\mathcal{E}_2(\mathcal{H}) = E(\mathcal{H}) \cap \mathcal{E}_2(\mathcal{G}_{r,\ell}(G))$. The underlying graph of $\mathcal{H}$ is denoted by $\mathbf{G}(\mathcal{H})$.

We find it instructive to first provide an informal overview of HYPERTREE. This algorithm takes a hypergraph $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$ as input, for some graph G on n vertices, builds a sequence $(\mathcal{H}_i)_{i=0}^T$ of subhypergraphs of $\mathcal{H}$ and outputs $\mathcal{H}_T$. The algorithm seeks to find a subhypergraph $\mathcal{F} \subseteq \mathcal{H}$ for which the following holds. The graph $F = \mathbf{G}(\mathcal{F})$, which is a subgraph of G, satisfies (1) $\lambda(F) \leq -\varepsilon$ or (2) $\lambda(F) \leq M$ and $e(F) \geq \log n$, for some positive constants $\varepsilon = \varepsilon(r, \ell)$ and $M = M(r, \ell)$.

Algorithm 1. HYPERTREE

Input: A hypergraph $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$, for some graph G with $V(G) = [n]$
Output: A pair $(\mathcal{H}_T, D_T)$, where $\mathcal{H}_T \subseteq \mathcal{H}$ and $D_T \subseteq \mathbb{N}$
/* Initialize: */
$i = 0$, $D_0 = \emptyset$, $\mathcal{H}_0 = \{E_0\}$ for some $E_0 \in \mathcal{E}_1(\mathcal{H})$
while $\lambda(\mathbf{G}(\mathcal{H}_i)) > -\varepsilon$ *and* $i < \log n$ **do**
  if *there exists* $E \in \mathcal{E}_1(\mathcal{H})$ *such that* $|V(E) \cap V(\mathbf{G}(\mathcal{H}_i))| \geq 2$ *and* $E \subsetneq V(\mathcal{H}_i)$ **then**
    set $\mathcal{H}_{i+1} = \mathcal{H}_i \cup \{E\}$ and $D_{i+1} = D_i \cup \{i+1\}$
  end
  else
    let $\mathcal{H}_F$ be the output of FLOWER$(\mathcal{H}_i, \mathcal{H})$
    set $\mathcal{H}_{i+1} = \mathcal{H}_i \cup \mathcal{H}_F$
    if $|V(\mathbf{G}(\mathcal{H}_{i+1})) \setminus V(\mathbf{G}(\mathcal{H}_i))| = (r-1)(\ell-1)-1$ **then** set $D_{i+1} = D_i$
    else set $D_{i+1} = D_i \cup \{i+1\}$
  end
  $i \mapsto i+1$
end
return $(\mathcal{H}_i, D_i)$

In the initialization step, the algorithm picks a hyperedge $E_0 \in \mathcal{E}_1(\mathcal{H})$ and sets $\mathcal{H}_0 = \{E_0\}$. Then, the algorithm enters a while loop. In iteration i of the loop, the algorithm attaches a hyperedge $E_i \in \mathcal{E}_1(\mathcal{H})$ to the current hypergraph $\mathcal{H}_{i-1}$ to build $\mathcal{H}_i$. It is required that such a copy intersects $\mathbf{G}(\mathcal{H}_{i-1})$ in at least two vertices, but is not a subgraph of $\mathbf{G}(\mathcal{H}_{i-1})$. If no such hyperedge exists, then the algorithm runs a subroutine which we call FLOWER. This algorithm, when called within HYPERTREE, returns (1) a hyperedge $C \in \mathcal{E}_2(\mathcal{H})$ which intersects $\mathcal{H}_{i-1}$ in at least one hypervertex and it is not contained in $\mathcal{H}_{i-1}$; and (2) a collection of hyperedges in $\mathcal{E}_1(\mathcal{H})$, each intersecting C in exactly one hypervertex. The output of FLOWER is attached to $\mathcal{H}_{i-1}$ to build $\mathcal{H}_i$. We defer the exact description of FLOWER until after the description of HYPERTREE.

A hyperedge E always corresponds to a set of edges of some underlying graph, and so we denote by $V(E)$ the set of vertices of $V(G)$ belonging to some edge in E, that is, $V(E) = \{v \in V(G) : \exists e \in E, v \in e\}$. For a hypergraph $\mathcal{H}$, $V(\mathcal{H})$ denotes the set $\cup_{E \in E(\mathcal{H})}\{e : e \in E\}$, and hence we have $V(\mathcal{H}) = E(\mathbf{G}(\mathcal{H}))$. We switch between these two equivalent perspectives throughout the algorithm and its analysis, whichever is more convenient at that point. Next is the formal description of the HYPERTREE algorithm. Recall that $\varepsilon = \varepsilon(r, \ell)$ is the small positive constant given by Lemma 3.3 (Algorithm 1).

The auxiliary set D_T in the output of HYPERTREE will help us to count all the possible outputs given by this algorithm. This set will also help us to ensure that $\mathbf{G}(\mathcal{H}_T)$ belongs to $\mathcal{J}_1 \cup \mathcal{J}_2$ (recall the definition of these sets in (1)). From now on, we say that the ith step of HYPERTREE$(\mathcal{H})$ is *degenerate* if $i \in D_T$, and *non-degenerate* otherwise.

Let us now turn to the subroutine FLOWER. The input of FLOWER is a tuple $(\mathcal{H}_i, \mathcal{H})$, where $\mathcal{H}$ is a $\star$-critical subhypergraph of $\mathcal{G}_{r,\ell}(G)$, for some graph G, and $\mathcal{H}_i \subseteq \mathcal{H}$. When called within HYPERTREE, the output is a subhypergraph of $\mathcal{H}$ called a flower. For a hyperedge C of type 2 and hyperedges $P_1, \ldots, P_t$ of type 1, we call the hypergraph $\mathcal{H}_F = \{C, P_1, \ldots, P_t\}$ a *flower* if $|C \cap P_s| = 1$ and $C \cap P_s \cap P_q = \emptyset$ for all $1 \leq s < q \leq t$. The hyperedges $P_1, \ldots, P_t$ are called *petals*. Observe that $\mathbf{G}(\mathcal{H}_F)$

corresponds to a copy of C_ℓ and t copies of K_r that intersect C in exactly one edge (and possibly more vertices). Moreover, if we denote the edges in C by $e_0, \dots, e_{\ell-1}$, then the condition $|V(\mathbf{G}(\mathcal{H}_{i+1})) \setminus V(\mathbf{G}(\mathcal{H}_i))| = (r-1)(\ell-1)-1$ in line 7 of HYPERTREE is equivalent to having $t = \ell - 1$ and having (up to a relabeling) $V(C) \cap V(\mathbf{G}(\mathcal{H}_i)) = e_0$, $V(P_s) \cap V(C) = e_s$ and $V(P_s) \cap (V(\mathbf{G}(\mathcal{H}_{i+1})) \setminus V(C)) = \emptyset$ for $1 \leq s < \ell$.

In Section 6, we prove that the if condition in line 7 of HYPERTREE is satisfied for all but a constant number of iterations of the while loop. That is, the size of D_T is bounded by a constant. This will be one main ingredient to show that the set of all possible outputs given by HYPERTREE, up to isomorphism, is at most polylogarithmic in n when applied over $\bigcup_{V(G)=[n]} \mathrm{Crit}_{r,\ell}(G)$.

The second main ingredient towards this goal is to make sure that in a non-degenerate step i, there is only one way, up to isomorphism, to attach the underlying graph of a flower $\mathcal{H}_F$ to the current graph $\mathbf{G}(\mathcal{H}_i)$, independent of the input hypergraph $\mathcal{H}$. To make this precise, we now introduce some non-standard notation. For each $i \in \mathbb{N}$ and each graph H with $V(H) \subseteq [n]$, let $\mathscr{C}(i, H)$ be the set of all hypergraphs $\mathcal{F}$ with the following properties: (1) $\mathcal{F} \in \mathrm{Crit}_{r,\ell}(G)$ for some graph G with $V(G) = [n]$; (2) $H \cong \mathbf{G}(\mathcal{F}_i)$, where $\mathcal{F}_i$ is the subhypergraph of $\mathcal{F}$ generated after i iterations of the while loop of HYPERTREE($\mathcal{F}$); and (3) HYPERTREE($\mathcal{F}$) enters FLOWER in iteration $i+1$ of the while loop. If $\mathcal{F} \in \mathscr{C}(i, H)$, then by (2) there exists a graph isomorphism $\sigma_{\mathcal{F}_i} : V(H) \to V(\mathbf{G}(\mathcal{F}_i))$ between H and $\mathbf{G}(\mathcal{F}_i)$, and we fix one such $\sigma_{\mathcal{F}_i}$ for every $\mathcal{F}$. Abusing of notation, we denote by $\sigma_{\mathcal{F}_i}(E)$ the set of edges in $\mathbf{G}(\mathcal{F}_i)$ corresponding to the edges in E under the isomorphism $\sigma_{\mathcal{F}_i}$. Now, define

$$\overline{\mathcal{H}}_{H,i} = \bigcup_{\mathcal{F} \in \mathscr{C}(i,H)} \left\{ E \subseteq E(H) : \sigma_{\mathcal{F}_i}(E) \in \mathcal{F}_i \right\}.$$

Observe that if $\mathscr{C}(i, H) \neq \emptyset$, then $H = \mathbf{G}(\overline{\mathcal{H}}_{H,i})$. Finally, we call an edge e of a graph H *suitable (with respect to H)* if $e \notin C$ for all $C \in \mathcal{E}_2(\overline{\mathcal{H}}_{H,i})$. Let $S = S(H)$ be the set of suitable edges of H. Our next lemma says that if $\mathscr{C}(i, H) \neq \emptyset$, then $S(H) \neq \emptyset$.

Lemma 5.1. *Let G be a graph on vertex set $[n]$ and let $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$. Suppose that the algorithm Flower($\mathcal{H}_i, \mathcal{H}$) is called in iteration $i+1$ of the while loop of Hypertree($\mathcal{H}$). Then $S(\mathbf{G}(\mathcal{H}_i))$ is non-empty.*

We prove the lemma after the description of FLOWER. Lemma 5.1 allows us to make the following definition.

Definition 5.2 (Canonical edge)**.** For every graph H with $V(H) \subseteq [n]$ and $S(H) \neq \emptyset$, fix an edge $e_0 = e_0(H) \in S(H)$ such that if H and H' are isomorphic then there is a graph isomorphism $\varphi : V(H) \to V(H')$ that maps $e_0(H)$ to $e_0(H')$. Call $e_0(H)$ the *canonical edge of H*.

We remark that the canonical edge is undefined when $S(H)$ is empty. However, we only need the canonical edge of H when $H \cong \mathbf{G}(\mathcal{H}_i)$ and FLOWER is called on input $(\mathcal{H}_i, \mathcal{H})$ in HYPERTREE. In this case, Lemma 5.1 guarantees the existence of the canonical edge. We stress that, in the light of bounding the size of $\mathrm{Out}_{r,\ell}(n)/_{\cong}$, it is important that "the same" edge is fixed for any two isomorphic graphs.

Now we are ready to state the formal description of FLOWER (Algorithm 2).

We remark here that the condition $C \nsubseteq V(\mathcal{H}_i)$ in line 2 is not an additional restriction. That is, if $e_0 \in S(\mathbf{G}(\mathcal{H}_i))$, then $C \nsubseteq V(\mathcal{H}_i)$ for all $C \in \mathcal{E}_2(\mathcal{H})$ containing e_0. Indeed, if we had $C \subseteq V(\mathcal{H}_i)$ for some $C \in \mathcal{E}_2(\mathcal{H})$ containing e_0, then this would imply that $C \in \mathcal{E}_2(\overline{\mathcal{H}}_{G_i,i})$, where $G_i = \mathbf{G}(\mathcal{H}_i)$ for simplicity. Thus, e_0 would not be suitable, a contradiction.

Algorithm 2. FLOWER

Input: A tuple $(\mathcal{H}_i, \mathcal{H})$, where $\mathcal{H}$ is a hypergraph in $\mathrm{Crit}_{r,\ell}(G)$, for some graph G, and $\mathcal{H}_i \subseteq \mathcal{H}$
Output: A flower $\mathcal{H}_F = \{C\} \cup \{P_e : e \in C \setminus V(\mathcal{H}_i)\}$, where $C \in \mathcal{E}_2(\mathcal{H})$, $P_e \in \mathcal{E}_1(\mathcal{H})$ for all $e \in C \setminus V(\mathcal{H}_i)$

```
/* Find a seed:                                              */
```

1 Let $e_0 \in S(\mathbf{G}(\mathcal{H}_i))$ be the canonical edge of $\mathbf{G}(\mathcal{H}_i)$
2 Let $C \in \mathcal{E}_2(\mathcal{H})$ be a hyperedge containing e_0 such that $C \nsubseteq V(\mathcal{H}_i)$
for *every* $e \in C \setminus V(\mathcal{H}_i)$ **do**
3 | let $P_e \in \mathcal{E}_1(\mathcal{H})$ be such that $C \cap P_e = \{e\}$
end
return $\{C\} \cup \{P_e : e \in C \setminus V(\mathcal{H}_i)\}$

Proof of Lemma 5.1. Let $G_i := \mathbf{G}(\mathcal{H}_i)$ and let $\overline{\mathcal{H}}_i := \overline{\mathcal{H}}_{G_i,i}$. Recall that we have $G_i = \mathbf{G}(\overline{\mathcal{H}}_i)$ as remarked after the definition of $\overline{\mathcal{H}}_i$. We first claim that $\overline{\mathcal{H}}_i$ cannot be ⋆-critical. Otherwise, $\lambda(G_i) \leq -\varepsilon$, by Lemma 3.3, and this would imply that HYPERTREE($\mathcal{H}$) has not entered the while loop in iteration $i+1$. In particular, FLOWER would not be called in iteration $i+1$ of HYPERTREE($\mathcal{H}$).

Now, we show that for every $C \in \mathcal{E}_2(\overline{\mathcal{H}}_i)$ and every $e \in C$, there exists $K \in \mathcal{E}_1(\overline{\mathcal{H}}_i)$ such that $K \cap C = \{e\}$. Indeed, let $C \in \mathcal{E}_2(\overline{\mathcal{H}}_i)$, and let $\mathcal{F} \in \mathscr{C}(i, G_i)$ be such that $\sigma_{\mathcal{F}_i}(C) \in \mathcal{E}_2(\mathcal{F})$, where $\mathcal{F}_i$ is the subhypergraph of $\mathcal{F}$ generated after i iterations of the while loop of HYPERTREE($\mathcal{F}$) and where $\sigma_{\mathcal{F}_i} : V(G_i) \to V(\mathbf{G}(\mathcal{F}_i))$ is a graph isomorphism. Note that such $\mathcal{F}$ must exist by definition of $\overline{\mathcal{H}}_i$. Let $e \in C$ be arbitrary and let e' be the corresponding copy of e in $\mathbf{G}(\mathcal{F}_i)$. Since $\mathcal{F}$ is ⋆-critical, there must exist $K' \in \mathcal{E}_1(\mathcal{F})$ such that $K' \cap \sigma_{\mathcal{F}_i}(C) = \{e'\}$. However, as HYPERTREE($\mathcal{F}$) has entered the else-statement in iteration $i+1$ (cf. the definition of $\mathscr{C}(i, G_i)$), the condition of the if statement in line 3 is false. This implies that $K' \subseteq V(\mathcal{F}_i)$, and hence the preimage K of K' under $\sigma_{\mathcal{F}_i}$ is contained in $\mathcal{E}_1(\overline{\mathcal{H}}_i)$. In particular, we have $K \cap C = \{e\}$.

It follows that the only reason for which $\overline{\mathcal{H}}_i$ is not ⋆-critical is because there exists an edge $e \in V(\overline{\mathcal{H}}_i) = E(G_i)$ for which there is no hyperedge in $\mathcal{E}_2(\overline{\mathcal{H}}_i)$ containing e. Or, equivalently, $S(G_i) \neq \emptyset$. ■

The following is now immediate.

Corollary 5.3. *Under the same assumptions as in Lemma 5.1, the algorithm Flower$(\mathcal{H}_i, \mathcal{H})$ runs without errors and finishes in finite time. Moreover, the edge e_0 in line 1 is uniquely determined by the isomorphism class of* $\mathbf{G}(\mathcal{H}_i)$.

Proof. The existence of the edge e_0 in line 1 follows immediately from $S(\mathbf{G}(\mathcal{H}_i)) \neq \emptyset$ proved in the previous lemma. Its uniqueness follows by fixing the canonical edge for every isomorphism type globally. As $\mathcal{H}$ is ⋆-critical, the existence of the cycle C as in line 2 and the petals P_e as in line 3 is straightforward. ■

We next pin down important properties of the flower returned by FLOWER$(\mathcal{H}_i, \mathcal{H})$ that we need repeatedly in the analysis of the algorithm.

Lemma 5.4. *Under the same assumptions as in Lemma 5.1, Flower $(\mathcal{H}_i, \mathcal{H})$ outputs a flower $\mathcal{H}_F = \{C\} \cup \{P_e : e \in C \setminus V(\mathcal{H}_i)\}$ which satisfies the following properties:*

(F1) $C \in \mathcal{E}_2(\mathcal{H})$ *but* $C \nsubseteq V(\mathcal{H}_i)$,

(F2) $P_e \in \mathcal{E}_1(\mathcal{H})$ *and* $C \cap P_e = \{e\}$ *for every* $e \in C \setminus V(\mathcal{H}_i)$*, and*
(F3) $|V(P_e) \cap V(\mathbf{G}(\mathcal{H}_i))| \leq 1 \leq |C \cap E(\mathbf{G}(\mathcal{H}_i))|$ *for all* $e \in C \setminus V(\mathcal{H}_i)$*.*

Proof. The properties (*F1*) and (*F2*) are immediate from the algorithm description and Corollary 5.3. As FLOWER was called in iteration $i + 1$ of the while loop of HYPERTREE($\mathcal{H}$), line 3 of HYPERTREE was not executed. This means that for each petal P_e we have $|V(P_e) \cap V(\mathbf{G}(\mathcal{H}_i))| \leq 1$, which proves the first inequality in (*F3*). The second inequality follows from the existence of e_0 in line 1, as $e_0 \in C \cap E(\mathbf{G}(\mathcal{H}_i))$. ■

With Corollary 5.3 and Lemma 5.4 at our hands, we now deduce Lemma 3.4.

Proof of Lemma 3.4. In its initialization, the algorithm HYPERTREE($\mathcal{H}$) sets $D_0 = \emptyset$ and $\mathcal{H}_0 = \{E_0\}$, for some $E_0 \in \mathcal{E}_1(\mathcal{H})$. This already establishes Part (*a*). Now, for each iteration $i + 1$ of the while loop, where $i = 0, 1, \ldots$, HYPERTREE($\mathcal{H}$) executes one of the following actions. Either it sets $\mathcal{H}_{i+1} = \mathcal{H}_i \cup \{E\}$ for some $E \in \mathcal{E}_1(\mathcal{H})$ (Case 1, cf. line 4), or it sets $\mathcal{H}_{i+1} = \mathcal{H}_i \cup \mathcal{H}_F$, where $\mathcal{H}_F$ is the output of the algorithm FLOWER($\mathcal{H}_i, \mathcal{H}$) (Case 2, cf. line 6). We remark that one of them must be executed because $\mathcal{H}_i \notin \mathrm{Crit}_{r,\ell}(G)$, as $\lambda(\mathbf{G}(\mathcal{H}_i)) > -\varepsilon$ (see Lemma 3.3). In either case, $\mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \mathcal{H}$ (cf. Lemma 5.4 for the second case). Similarly, it is easy to see from lines 4, 7, and 8 that $D_i \subseteq D_{i+1} \subseteq \mathbb{N}$.

Let T be the number of iterations of the while loop in line 2 of HYPERTREE($\mathcal{H}$). By the while loop condition and the increase of i by one in every iteration (see line 9), HYPERTREE($\mathcal{H}$) must stop in at most $\log n$ iterations. Moreover, the while loop guarantees that T is the smallest integer such that $\lambda(G_T) \leq -\varepsilon$ or $T \geq \log n$, where $G_T = \mathbf{G}(\mathcal{H}_T)$ and ε is the constant given by Lemma 3.3. This establishes Part (*c*). Since HYPERTREE($\mathcal{H}$) should return $(\mathcal{H}_T, D_T)$ (see line 10), we also establish Part (*d*).

Now, it remains to show Part (*b*). Let $i = 0, 1, \ldots$, and assume HYPERTREE($\mathcal{H}$) enters the while loop in iteration $i+1$. In Case 1, the hyperedge E satisfies $E \nsubseteq V(\mathcal{H}_i)$ (cf. line 3). In Case 2, Lemma 5.4 implies that the output $\mathcal{H}_F = \{C, P_1, \ldots, P_t\}$ given by FLOWER($\mathcal{H}_i, \mathcal{H}$) satisfies $C \nsubseteq V(\mathcal{H}_i)$. In both cases, $v(\mathcal{H}_i) < v(\mathcal{H}_{i+1})$. ■

6 | THE ALGORITHM ANALYSIS

In this section, we prove Lemmas 3.5 and 3.6, and hence complete the proof of Theorem 2.1. Let G_i denote the graph $\mathbf{G}(\mathcal{H}_i)$, where $\mathcal{H}_i$ is the hypergraph generated in the ith step of HYPERTREE. Lemma 3.5 is easily deduced from our next lemma, which states that $\lambda(\mathbf{G}(\mathcal{H}_i))$ either decreases by an additive constant in a degenerate step or it remains the same in a non-degenerate step. This lemma is proved at the end of this section.

Lemma 6.1. *For all integers* $r, \ell \geq 4$*, there exists* $\delta = \delta(r, \ell) > 0$ *such that the following holds. For any graph* G *and any hypergraph* $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(G)$*, the sequence* $(\mathcal{H}_i, D_i)_{i=0}^T$ *generated by Hypertree*($\mathcal{H}$) *satisfies*

(1) $\lambda(G_i) = \lambda(G_{i-1})$ *for all* $i \notin D_T$*, and*
(2) $\lambda(G_i) \leq \lambda(G_{i-1}) - \delta$ *for all* $i \in D_T$*,*

where $G_i = \mathbf{G}(\mathcal{H}_i)$ *for each* $i \in \{1, \ldots, T\}$*.*

For all $1 \leq i \leq T$, the graph G_i is obtained from G_{i-1} by adding either an r-clique or the underlying graph of a flower $\{C, P_1, \ldots, P_t\}$ to it, depending whether HYPERTREE executes the if-clause in lines

3–4 or the else-clause in lines 5–8. In the latter case, we will analyze the change in λ by adding first C, and then one petal (copy of K_r) at a time. Thus, it makes sense to pin down the effect of adding a copy of K_r to an arbitrary graph F first.

For two graphs F_1 and F_2, we denote by $F_1 \cap F_2$ the subgraph with vertex set $V(F_1) \cap V(F_2)$ and edge set $E(F_1) \cap E(F_2)$. The graph $F_1 \cup F_2$ is defined analogously. For any graph F, recall that $\lambda(F) = v(F) - e(F)/m_2(K_r, C_\ell)$. Then, we can write

$$\begin{aligned}\lambda(F_1 \cup F_2) - \lambda(F_1) &= v(F_1 \cup F_2) - v(F_1) - \frac{e(F_1 \cup F_2) - e(F_1)}{m_2(K_r, C_\ell)}\\ &= v(F_2) - v(F_1 \cap F_2) - \frac{e(F_2) - e(F_1 \cap F_2)}{m_2(K_r, C_\ell)}.\end{aligned} \tag{9}$$

Now, define

$$\beta_{r,\ell}(J) = r - v(J) - \frac{\binom{r}{2} - e(J)}{m_2(K_r, C_\ell)}. \tag{10}$$

By (9), we have

$$\lambda(F_1 \cup F_2) - \lambda(F_1) = \beta_{r,\ell}(J), \tag{11}$$

in the case when $F_2 \cong K_r$ and $J = F_1 \cap F_2$. Before stating the lemma which encompasses how $\beta_{r,\ell}(J)$ behaves for various subgraphs $J \subseteq K_r$, we pin down the following fact which provides closed formulas for $m_2(C_\ell)$, $m_2(K_r)$, and $m_2(K_r, C_\ell)$. These follow from standard calculations which we provide in the Appendix for completeness.

Fact 6.2. Let $r, \ell \geq 4$ be integers. Then,

$$m_2(C_\ell) = \frac{\ell - 1}{\ell - 2}, \quad m_2(K_r) = \frac{r+1}{2}, \quad \text{and} \quad m_2(K_r, C_\ell) = \frac{\binom{r}{2}}{r - 2 + (\ell-2)/(\ell-1)}.$$

In particular, $r/2 < m_2(K_r, C_\ell) < m_2(K_r)$.

In our next lemma, we obtain upper bounds for $\beta_{r,\ell}(J)$ for every subgraph $J \subsetneq K_r$ with at least two vertices.

Lemma 6.3. *Let $r, \ell \geq 4$ be integers. Let $J \subsetneq K_r$ such that $v(J) \geq 2$. Then,*

(a) $\beta_{r,\ell}(J) < 0$,
(b) $\beta_{r,\ell}(K_2) = 1/m_2(K_r, C_\ell) - (\ell-2)/(\ell-1) > -1$,
(c) $\beta_{r,\ell}(J) \leq \beta_{r,\ell}(K_2)$ *if* $d(v) = 1$ *for some* $v \in V(J)$. *The equality holds if and only if* $J \cong K_2$.

Proof. First, let us prove part (a). When $J \subsetneq K_r$ has r vertices, we can easily see from (10) that $\beta_{r,\ell}(J) < 0$. Thus, let us assume that $2 \leq v(J) < r$. Observe that $\beta_{r,\ell}(J) < 0$ if the following inequalities are satisfied:

$$m_2(K_r, C_\ell) < m_2(K_r) \leq \frac{\binom{r}{2} - e(J)}{r - v(J)}. \tag{12}$$

The first inequality follows from Fact 6.2. For the second, simply note that

$$\frac{\binom{r}{2}-e(J)}{r-v(J)} \geq \frac{\binom{r}{2}-\binom{v(J)}{2}}{r-v(J)} \geq \frac{r+j-1}{2} \geq \frac{r+1}{2}.$$

As $m_2(K_r) = (r+1)/2$ (see Fact 6.2), this establishes part (a).

To show part (c), first note that $d(v) = 1$ for some $v \in V(J)$ if and only if $K_2 \subseteq J \subseteq K_{r-1} \cdot K_2$, where $K_{r-1} \cdot K_2$ denotes the graph obtained from K_{r-1} by adding a pendant edge. When $J \cong K_2$, the equality in (c) holds trivially. Thus, let us assume that $v(J) \geq 3$ and $J \subseteq K_{r-1} \cdot K_2$. In this case, the inequality $\beta_{r,\ell}(J) < \beta_{r,\ell}(K_2)$ is equivalent to

$$\frac{e(J)-1}{v(J)-2} < m_2(K_r, C_\ell). \tag{13}$$

But, for any $J \subseteq K_{r-1} \cdot K_2$ such that $v(J) \geq 3$, we have

$$\frac{e(J)-1}{v(J)-2} \leq m_2(K_{r-1} \cdot K_2) = \max\left\{ m_2(K_{r-1}), \frac{e(K_{r-1} \cdot K_2)-1}{v(K_{r-1} \cdot K_2)-2} \right\} = \frac{r}{2},$$

by definition of $m_2(\cdot)$ and the identity $m_2(K_{r-1}) = r/2$ (see Fact 6.2). As $m_2(K_r, C_\ell) > r/2$ by Fact 6.2, this finishes the proof of part (c).

For part (b), the identity $\beta_{r,\ell}(K_2) = 1/m_2(K_r, C_\ell) - (\ell-2)/(\ell-1)$ follows readily from the definition of $\beta_{r,\ell}$ in (10) and the identity for $m_2(K_r, C_\ell)$ in Fact 6.2. Finally, $m_2(K_r, C_\ell) > 0$ and $(\ell-2)/(\ell-1) < 1$ imply that $\beta_{r,\ell}(K_2) > -1$. ■

Now we are ready to prove Lemma 6.1. As a consequence, we also prove Lemma 3.5.

Proof of Lemma 6.1. Suppose that HYPERTREE($\mathcal{H}$) executes the if-statement in lines 3–4 in the ith iteration of its while loop. Then, $i \in D_i$ and hence $i \in D_T$. Moreover, $G_i = G_{i-1} \cup K$ for some $K \cong K_r$ such that $|V(G_{i-1}) \cap V(K)| \geq 2$ and $K \nsubseteq G_{i-1}$. Observe that the graph $J = G_{i-1} \cap K$ satisfies the assumptions of Lemma 6.3 and hence, by (11), $\lambda(G_i) - \lambda(G_{i-1}) = \beta_{r,\ell}(J) < 0$.

Now, suppose that HYPERTREE($\mathcal{H}$) executes the else-statement in lines 5–8 in the ith iteration of its while loop. Let $\mathcal{H}_F = \{C\} \cup \{P_e : e \in C \setminus E(G_{i-1})\}$ be the flower returned by FLOWER($\mathcal{H}_{i-1}, \mathcal{H}$). Recall all the properties of $\mathcal{H}_F$ given by Lemma 5.4. In order to bound the difference $\lambda(G_i) - \lambda(G_{i-1})$, we first analyze the increment $\lambda(G_{i-1} \cup C) - \lambda(G_{i-1})$. Let J_0 be the graph $G_{i-1} \cap C$. By (9), we have

$$\begin{aligned}\lambda(G_{i-1} \cup C) - \lambda(G_{i-1}) &= \ell - v(J_0) - \frac{\ell - e(J_0)}{m_2(K_r, C_\ell)} \\ &\leq \left(\frac{\ell-2}{\ell-1} - \frac{1}{m_2(K_r, C_\ell)}\right) \cdot |E(C) \setminus E(G_{i-1})|,\end{aligned} \tag{14}$$

where in the inequality we use that $v(J_0) \geq e(J_0) + 1 \geq 2$, as $K_2 \subseteq J_0 \subsetneq C_\ell$ (see Lemma 5.4). Note that equality holds in (14) if and only if $J_0 \cong K_2$.

By Lemma 6.3(a), the contribution of each petal of $\mathcal{H}_F$ to λ is negative. But, the contribution of C to λ, which is bounded by (14), may be positive (and large). However, as we shall show, the contribution of C to λ is smaller than or equal to the absolute value of the sum of all the contributions of each petal of $\mathcal{H}_F$ to λ. In order to prove this, we recursively find a subsequence of petals $(P_j)_{j=1}^t$ in $\mathcal{H}_F$ such that the intersection graph $P_j \cap (G_{i-1} \cup C \cup P_1 \cup \cdots \cup P_{j-1})$ has potentially many isolated vertices.

These isolated vertices allow us to gain a sufficiently negative contribution to λ from each petal in the sequence, and hence "beat" the contribution given by the cycle in (14). This sequence of petals does not necessarily contain all the petals of $\mathcal{H}_F$, but this is not a problem. By Lemma 6.3(a), all the petals in $\mathcal{H}_F$ give a negative contribution to λ, and hence we may discard some petals from the analysis (and adding them later will not increase the value of λ).

We define this sequence of petals iteratively. Let us say that $V(C) = \{u_0, \dots u_{\ell-1}\}$ with $u_{j-1}u_j \in E(C)$, for each $j \in [\ell]$ (assuming that $u_\ell = u_0$), and that the edge $u_0u_{\ell-1}$ belongs to G_{i-1}. Now, define $A_0 = E(C) \setminus E(G_{i-1})$ and construct a nested sequence of sets $(A_s)_{s\geq 0}$ in the following recursive way. For each $s \in \mathbb{N}$, if A_{s-1} is empty, then let $A_s = \emptyset$. If A_{s-1} is non-empty, then let

$$m_s = \min\{m : u_m u_{m+1} \in A_{s-1}\},$$

and let $P_s = P_{u_{m_s}u_{m_s+1}}$ be the petal in $\mathcal{H}_F$ which covers the edge $u_{m_s}u_{m_s+1}$. Then, set

$$A_s = A_{s-1} \setminus \{u_m u_{m+1} : u_{m+1} \in V(P_s)\}.$$

Let t be the smallest integer such that $A_t = \emptyset$, and note that $t \leq |A_0| = |E(C) \setminus E(G_{i-1})|$.

For simplicity, denote $G_{i-1}^{(0)} = G_{i-1} \cup C$ and, more generally, for each $1 \leq s \leq t$, let

$$G_{i-1}^{(s)} = G_{i-1} \cup C \cup P_1 \cup \cdots \cup P_s.$$

Now, observe that

$$\lambda(G_i) - \lambda(G_{i-1}) \leq \lambda(G_{i-1}^{(0)}) - \lambda(G_{i-1}) + \sum_{s=1}^{t} \left(\lambda(G_{i-1}^{(s)}) - \lambda(G_{i-1}^{(s-1)})\right). \tag{15}$$

Indeed, Lemma 6.3(a) together with (11) imply that we can discard the petals in $\{P_e : e \in E(C) \setminus E(G_{i-1})\}$ which do not belong to the chosen sequence $P_1, \dots, P_t$. Moreover, equality holds if and only if $P_1, \dots, P_t$ are all the petals in the flower $\mathcal{H}_F$. To bound each increment in (15), we next analyze the structure of the graph $J_s := G_{i-1}^{(s-1)} \cap P_s$. Define

$$I_s = \{u_{m+1} \in V(J_s) \setminus \{u_{m_s+1}\} : u_m u_{m+1} \in A_{s-1}\}.$$

Claim 6.4. The degree of u_{m_s+1} in J_s is 1 and I_s is a set of isolated vertices in J_s.

Proof. Let u_m be any vertex in $I_s \cup \{u_{m_s+1}\}$ and w be any vertex in J_s. We affirm that u_m is adjacent to w inside the graph J_s if and only if $\{u_m w\} = C \cap P_s$. Indeed, we cannot have $u_m w \in E(G_{i-1})$, otherwise P_s would be an r-clique which intersects G_{i-1} in at least 2 vertices, contradicting Lemma 5.4 ($F3$). If $w \neq u_{m+1}$, we also cannot have $\{u_m w\} \in E(P_1 \cup \cdots \cup P_{s-1})$, otherwise $u_{m-1}u_m \notin A_{s-1}$, and hence $u_m \notin I_s \cup \{u_{m_s+1}\}$.

As $u_{m_s}u_{m_s+1}$ is the only edge in $P_s \cap C$, it follows that u_{m_s} is the only neighbor of u_{m_s+1} in J_s, and that u_m is isolated in J_s for any $u_m \in I_s$. ■

Let $\widetilde{J}_s$ be the subgraph of J_s induced by the vertex set $V(J_s) \setminus I_s$. By the previous claim, we have $E(\widetilde{J}_s) = E(J_s)$, which implies that $\beta_{r,\ell}(J_s) = \beta_{r,\ell}(\widetilde{J}_s) - |I_s|$ (see (10)). By (11), we obtain

$$\lambda(G_{i-1}^{(s)}) - \lambda(G_{i-1}^{(s-1)}) = \beta_{r,\ell}(J_s) = \beta_{r,\ell}(\widetilde{J}_s) - |I_s| \leq \beta_{r,\ell}(K_2) - |I_s| \tag{16}$$

for every $1 \le s \le t$. In the last inequality we use Lemma 6.3(*c*), as $d(u_{m_s+1}) = 1$ by Claim 6.4. Moreover, by Lemma 6.3(*c*), equality holds if and only if $\widetilde{J}_s \cong K_2$. When $\widetilde{J}_s \cong K_2$, note that we also have $|I_s| = 0$, as the only vertex $u_{m+1} \in V(J_s)$ such that $u_m u_{m+1} \in A_{s-1}$ is $u_{m+1} = u_{m_s+1}$.

Combining (14)–(16), we have

$$\lambda(G_i) - \lambda(G_{i-1}) \le \left(\frac{\ell-2}{\ell-1} - \frac{1}{m_2(K_r, C_\ell)}\right) \cdot |E(C) \setminus E(G_{i-1})| + t\beta_{r,\ell}(K_2) - \sum_{s=1}^{t} |I_s|. \tag{17}$$

From the definitions of A_s and I_s, it is easy to see that $\sum_s (|I_s| + 1) = |A_0| = |E(C) \setminus E(G_{i-1})|$. And, by Lemma 6.3(*b*), we have $\beta_{r,\ell}(K_2) = m_2(K_r, C_\ell)^{-1} - (\ell-2)/(\ell-1)$. Then, (17) is equivalent to

$$\lambda(G_i) - \lambda(G_{i-1}) \le (\beta_{r,\ell}(K_2) + 1) \cdot (t - |A_0|). \tag{18}$$

By Lemma 6.3, $\beta_{r,\ell}(K_2) > -1$ and, as we have $t \le |A_0|$, it follows that

$$(\beta_{r,\ell}(K_2) + 1) \cdot (t - |A_0|) \le 0. \tag{19}$$

Clearly, equality in (19) holds if and only if $t = |A_0|$. We conclude that $\lambda(G_i) - \lambda(G_{i-1}) \le 0$ in the case when we add the flower $\{C\} \cup \{P_e : e \in C \setminus E(G_{i-1})\}$.

Observe that $\lambda(G_i) - \lambda(G_{i-1}) = 0$ if and only if we have equalities in (14)–(19). This means that we must have $C \cap G_{i-1} \cong K_2$ (and hence $|A_0| = \ell - 1$), $t = |A_0|$ and

$$P_s \cap (G_{i-1} \cup C \cup P_1 \cup \cdots \cup P_{s-1}) \cong K_2,$$

for each $1 \le s \le t$. As $e \in E(P_e \cap C)$, we infer that none of the $\ell - 1$ petals intersect outside the cycle C and that the only petals sharing a vertex are consecutive petals, which share exactly one vertex. This happens if and only if $|V(G_i) \setminus V(G_{i-1})| = (r-1)(\ell-1)-1$, in which case i is *not* added to D_i (cf. line 7 of HYPERTREE), and then $i \notin D_T$. This proves (1). The existence of $\delta = \delta(r, \ell)$ for (2) readily follows by noting that there are only $C = C(r, \ell)$ non-isomorphic configurations of such flowers and cliques (and how they intersect with G_{i-1}). This finishes the proof of the lemma. ■

It remains to prove Lemma 3.6, which bounds the number of non-isomorphic underlying graphs that HYPERTREE may output. Recall that, for a set of graphs S, we denote by $S/_{\cong}$ a set consisting of exactly one representative graph for every (graph) isomorphism class of S. In principle, $|\mathrm{Out}_{r,\ell}(n)/_{\cong}|$ could be very large, but this is avoided by two means. First, the number of degenerate steps in HYPERTREE is bounded by a constant. In this case, we bound the number of possible (non-isomorphic) structures that can emerge in one iteration of the while loop of HYPERTREE quite crudely. Second, in a non-degenerate step, we ensure that there is only one possible structure emerging from a given $\mathbf{G}(\mathcal{H}_t)$, up to isomorphism. Here, it is important that we fix the canonical edge of $\mathbf{G}(\mathcal{H}_t)$ in a unique way for the class of graphs isomorphic to $\mathbf{G}(\mathcal{H}_t)$, compare Definition 5.2.

We first bound how many non-isomorphic graphs $G_t = \mathbf{G}(\mathcal{H}_t)$ the algorithm HYPERTREE can produce in step t, for all $t = 1, \ldots, \lceil \log n \rceil$. To do so, we need to recall and define some notation. For each $n \in \mathbb{N}$, recall that $\mathrm{Crit}_{r,\ell}(n) = \bigcup_{V(G)=[n]} \mathrm{Crit}_{r,\ell}(G)$. For a hypergraph $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(n)$, let $T(\mathcal{H})$ be the stopping time of HYPERTREE($\mathcal{H}$). For any $t \ge 0$ and any $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(n)$ such that $T(\mathcal{H}) \ge t$, let $\mathcal{H}_t(\mathcal{H})$ be the hypergraph obtained in step t of HYPERTREE($\mathcal{H}$) and let $D_t(\mathcal{H})$ be the accompanying set of integers. Recall that $\mathcal{H}_t$ is a subgraph of $\mathcal{G}_{r,\ell}(G)$ for some graph G with $V(G) = [n]$, so that

$V(\mathcal{H}_t) \subseteq \binom{[n]}{2}$ and we can associate with $\mathcal{H}_t$ a graph $\mathbf{G}(\mathcal{H}_t)$, called the underlying graph of $\mathcal{H}_t$, which is a subgraph of G. Finally, for each $t, n \in \mathbb{N}$ and each set $D \subseteq \{1, \dots, t\}$, define

$$\mathcal{G}(t, D, n) = \bigcup \{\mathbf{G}(\mathcal{H}_t) : \mathcal{H}_t = \mathcal{H}_t(\mathcal{H})\},$$

where the union is over all $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(n)$ such that $D_t(\mathcal{H}) = D$ and $T(\mathcal{H}) \geq t$. Our next lemma gives an upper bound on the size of $\mathcal{G}(t, D, n)/_{\cong}$.

Lemma 6.5. *For all $r, \ell \geq 4$ there exists $C > 0$ such that $|\mathcal{G}(t, D, n)/_{\cong}| \leq (tr\ell)^{C|D|}$, for all $t, n \in \mathbb{N}$ and $D \subseteq \{1, \dots, t\}$.*

Proof. To simplify notation, set $g(t, D, n) := |\mathcal{G}(t, D, n)/_{\cong}|$. First, note that $\mathcal{G}(0, \emptyset, n)$ contains only one graph G_0, up to isomorphism, and hence $g(0, \emptyset, n) = 1$. Indeed, for every $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(n)$, the hypergraph $\mathcal{H}_0(\mathcal{H})$ consists of one hyperedge of type 1 (cf. Lemma 3.4(a)). That is, G_0 is a copy of K_r.

Now, we claim that for each $t \geq 1$ and each $D \subseteq \{1, \dots, t\}$, we have

$$g(t, D, n) \leq \begin{cases} g(t-1, D, n) & \text{if } t \notin D; \\ g(t-1, D \setminus \{t\}, n) \cdot (t\ell r)^{4(\ell r)^2} & \text{if } t \in D. \end{cases} \tag{20}$$

First, assume that $t \notin D$. Let $G_{t-1} \in \mathcal{G}(t-1, D, n)$. We will show that there is at most one graph $G_t \in \mathcal{G}(t, D, n)$, up to isomorphism, such that if $\mathcal{H}$ is a subgraph with $\mathbf{G}(\mathcal{H}_{t-1}(\mathcal{H})) \cong G_{t-1}$ and $T(\mathcal{H}) \geq t$ then we must have that $\mathbf{G}(\mathcal{H}_t(\mathcal{H})) \cong G_t$. If $t \geq \log n$ or $\lambda(G_{t-1}) \leq -\varepsilon$, then $T(\mathcal{H}) = t - 1$ for all $\mathcal{H}$ such that $\mathbf{G}(\mathcal{H}_{t-1}(\mathcal{H})) \cong G_{t-1}$. In this case, the statement is trivially true. So we may assume that $t \leq \log n$ and $\lambda(G_{t-1}) > -\varepsilon$. Let $\mathcal{H}$ be any hypergraph in $\mathrm{Crit}_{r,\ell}(n)$ such that $D_t(\mathcal{H}) = D$ and $\mathbf{G}(\mathcal{H}_{t-1}) \cong G_{t-1}$, where $\mathcal{H}_{t-1} = \mathcal{H}_{t-1}(\mathcal{H})$. As $t \notin D$ and HYPERTREE($\mathcal{H}$) did not stop after iteration t, we have $\mathcal{H}_t := \mathcal{H}_t(\mathcal{H}) = \mathcal{H}_{t-1} \cup \mathcal{H}_F$, for some flower $\mathcal{H}_F$ such that

$$|V(\mathbf{G}(\mathcal{H}_t)) \setminus V(\mathbf{G}(\mathcal{H}_{t-1}))| = (r-1)(\ell-1) - 1, \tag{21}$$

see line 7 of HYPERTREE. For (21) to hold, observe that $\mathbf{G}(\mathcal{H}_F)$ must intersect $\mathbf{G}(\mathcal{H}_{t-1})$ in exactly one edge, the canonical edge e_0 of $\mathbf{G}(\mathcal{H}_{t-1})$, see line 1 of *Flower* $(\mathcal{H}_{t-1}, \mathcal{H})$. Once we have this edge, we can see that FLOWER generates only one type of flower $\mathbf{G}(\mathcal{H}_F)$ such that (21) holds and $\mathbf{G}(\mathcal{H}_F) \cap \mathbf{G}(\mathcal{H}_{t-1})$ is equal to $\{e_0\}$. Moreover, by construction, e_0 only depends on the isomorphism class of $\mathbf{G}(\mathcal{H}_{t-1})$, see Definition 5.2 and Corollary 5.3. Therefore, for any other hypergraph $\widetilde{\mathcal{H}}$ such that $\mathbf{G}(\mathcal{H}_{t-1}(\widetilde{\mathcal{H}})) \cong \mathbf{G}(\mathcal{H}_{t-1}) \cong G_{t-1}$ and $t \notin D_t(\widetilde{\mathcal{H}})$, the flower $\widetilde{\mathcal{H}}_F$ given by the algorithm *Flower* in iteration t of HYPERTREE($\widetilde{\mathcal{H}}$) satisfies $\mathbf{G}(\widetilde{\mathcal{H}}_F) \cap \mathbf{G}(\mathcal{H}_{t-1}(\widetilde{\mathcal{H}})) = \{e'_0\}$, where e'_0 is the image of e_0 under some graph isomorphism $\varphi : V(\mathbf{G}(\mathcal{H}_{t-1})) \to V(\mathbf{G}(\mathcal{H}_{t-1}(\widetilde{\mathcal{H}})))$. Thus, we conclude that the two graphs $\mathbf{G}(\mathcal{H}_t(\mathcal{H}))$ and $\mathbf{G}(\mathcal{H}_t(\widetilde{\mathcal{H}}))$ produced after t iterations of the while loop by HYPERTREE($\mathcal{H}$) and HYPERTREE($\widetilde{\mathcal{H}}$), respectively, are isomorphic. This implies that $g(t, D, n) \leq g(t-1, D, n)$ when $t \notin D$, which proves the first inequality in (20).

Now, suppose that $t \in D$. To show the second inequality in (20), note that in step t of HYPERTREE($\mathcal{H}$) one of the following holds: (1) $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{E\}$, for some $E \in \mathcal{E}_1(\mathcal{H})$; or (2) $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \mathcal{H}_F$, for some flower $\mathcal{H}_F \subseteq \mathcal{H}$. Let $H = \mathbf{G}(E)$ or $H = \mathbf{G}(\mathcal{H}_F)$ be the underlying graph of the hyperedges that were added in step t. In order to count how many choices we have for the graph $\mathbf{G}(\mathcal{H}_{t-1}) \cup H$ it suffices to count how many subgraphs in $\mathbf{G}(\mathcal{H}_{t-1})$ have at most $v(H)$ vertices and how many subgraphs H has. As $v(H) \leq \ell r$ and $v(\mathbf{G}(\mathcal{H}_{t-1})) \leq t\ell r$, there are at most $(t\ell r)^{\ell r} \cdot 2^{(\ell r)^2}$ subgraphs in $\mathbf{G}(\mathcal{H}_{t-1})$ with at most ℓr vertices. Moreover, we can easily see that there are at most $(\ell r)^{\ell r} \cdot 2^{(\ell r)^2}$ subgraphs in H. From these bounds it follows that there are at most $(t\ell r)^{\ell r} 2^{(\ell r)^2} \cdot (\ell r)^{\ell r} 2^{(\ell r)^2} \leq (t\ell r)^{4(\ell r)^2}$

choices for the graph $\mathbf{G}(\mathcal{H}_{t-1}) \cup H$. That is, the graph $\mathbf{G}(\mathcal{H}_t)$ may be obtained from $\mathbf{G}(\mathcal{H}_{t-1})$ in at most $(t\ell r)^{4(\ell r)^2}$ ways, and hence $g(t, D, n) \leq g(t-1, D, n) \cdot (t\ell r)^{4(\ell r)^2}$.

As $g(0, \emptyset, n) = 1$, it follows that $g(t, D, n) \leq (t\ell r)^{4(\ell r)^2 |D|}$ by iterating the inequalities in (20). ∎

Now, we are ready to prove Lemma 3.6:

Proof of Lemma 3.6. We first claim that there exists a constant $C_1 = C_1(r, \ell) > 0$ such that $|D_T(\mathcal{H})| \leq C_1$ for all $\mathcal{H} \in \mathrm{Crit}_{r,\ell}(n)$. Recall that $T = T(\mathcal{H})$ denotes the stopping time of HYPERTREE($\mathcal{H}$). Fix any hypergraph $\mathcal{H}$ in $\mathrm{Crit}_{r,\ell}(n)$ and let $G_i = \mathbf{G}(\mathcal{H}_i)$ for $i = 0, \dots, T$. By Lemma 6.1, we have $\lambda(G_i) \leq \lambda(G_{i-1}) - \delta$ if $i \in D_T(\mathcal{H})$, and $\lambda(G_i) = \lambda(G_{i-1})$ if $i \notin D_T(\mathcal{H})$, where $\delta = \delta(r, \ell) > 0$. As $\lambda(G_0) = \lambda(K_r)$ (by Lemma 3.4(a)) and $\lambda(G_{T(\mathcal{H})-1}) > -\varepsilon$ (by Lemma 3.4(c)), it follows that $|D_T(\mathcal{H})| \leq 1 + (\lambda(K_r) + \varepsilon)/\delta$. As ε only depends on r and ℓ, this proves our claim.

By Lemma 3.4(c), the stopping time T is bounded from above by $\log n$. Since $|D_T| \leq C_1$, the size of $\mathrm{Out}_{r,\ell}(n)/_{\cong}$ is bounded by the size of

$$\bigcup_{t \leq \log n} \bigcup_{\substack{D \subseteq [t]: \\ |D| \leq C_1}} |\mathcal{G}(t, D, n)/_{\cong}|,$$

where $\mathcal{G}(t, D, n)$ was defined just above Lemma 6.5. Using the bound on $|\mathcal{G}(t, D, n)/_{\cong}|$ given by Lemma 6.5, we conclude that

$$|\mathrm{Out}_{r,\ell}(n)/_{\cong}| \leq \sum_{t=1}^{\lceil \log n \rceil} \sum_{\substack{D \subseteq [t]: \\ |D| \leq C_1}} (t\ell r)^{C|D|} \leq (\log n)^{C_0},$$

for some $C_0 = C_0(r, \ell) > 0$. ∎

ACKNOWLEDGMENTS

This work was started at the thematic program GRAPHS@IMPA (January–March 2018), in Rio de Janeiro. We thank IMPA and the organisers for the hospitality and for providing a pleasant research environment. We thank Rob Morris for helpful discussions, and the anonymous referees for their careful reading and many helpful suggestions. Open Access funding enabled and organized by Projekt DEAL.

FUNDING INFORMATION

A. Liebenau was supported by an ARC DECRA Fellowship Grant DE170100789. L. Mattos was supported by CAPES and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689). W. Mendonça was supported by CAPES project 88882.332408/2010-01

REFERENCES

1. J. Balogh, R. Morris, and W. Samotij, *Independent sets in hypergraphs*, J. Am. Math. Soc. **28** (2015), 669–709.
2. P. Frankl and V. Rödl, *Large triangle-free subgraphs in graphs without K_4*, Graphs Comb. **2** (1986), 135–144.

3. E. Friedgut and M. Krivelevich, *Sharp thresholds for certain Ramsey properties of random graphs*, Random Struct. Algorithms **17** (2000), 1–19.
4. L. Gugelmann, R. Nenadov, Y. Person, N. Škorić, A. Steger, and H. Thomas, *Symmetric and asymmetric Ramsey properties in random hypergraphs*, Forum Math. Sigma **5** (2017), e28.
5. Y. Kohayakawa and B. Kreuter, *Threshold functions for asymmetric Ramsey properties involving cycles*, Random Struct. Algorithms **11** (1997), 245–276.
6. Y. Kohayakawa, M. Schacht, and R. Spöhel, *Upper bounds on probability thresholds for asymmetric Ramsey properties*, Random Struct. Algorithms **44** (2014), 1–28.
7. B. Kreuter, *Threshold functions for asymmetric Ramsey properties with respect to vertex colorings*, Random Struct. Algorithms **9** (1996), 335–348.
8. T. Łuczak, A. Ruciński, and B. Voigt, *Ramsey properties of random graphs*, J. Comb. Theory Ser. B **56** (1992), 55–68.
9. M. Marciniszyn, J. Skokan, R. Spöhel, and A. Steger, *Asymmetric Ramsey properties of random graphs involving cliques*, Random Struct. Algorithms **34** (2009), 419–453.
10. F. Mousset, R. Nenadov, and W. Samotij, *Towards the Kohayakawa–Kreuter conjecture on asymmetric Ramsey properties*, Comb. Probab. Comput. (2020), 1–13.
11. F. Ramsey, *On a problem of formal logic*, Proc. Lond. Math. Soc. **30** (1930), 264–286.
12. V. Rödl and A. Ruciński, *Lower bounds on probability thresholds for Ramsey properties*, Combinatorics, Paul Erdős is eighty, Vol **1**, Bolyai Society Mathematical Studies, János Bolyai Mathematical Society, Budapest, 1993, pp. 317–346.
13. V. Rödl and A. Ruciński, *Threshold functions for Ramsey properties*, J. Am. Math. Soc. **8** (1995), 917–942.
14. D. Saxton and A. Thomason, *Hypergraph containers*, Invent. Math. **201** (2015), 925–992.

How to cite this article: A. Liebenau, L. Mattos, W. Mendonça, and J. Skokan, *Asymmetric Ramsey properties of random graphs involving cliques and cycles*, Random Struct. Alg. **62** (2023), 1035–1055. https://doi.org/10.1002/rsa.21106

APPENDIX A: PROOF OF FACT 6.2

Note that every subgraph $J \subsetneq C_\ell$ is a forest, and so we have $e(J) \leq v(J) - 1$. Thus, for every $J \subsetneq C_\ell$ with $v(J) \geq 3$ this implies that $(e(J) - 1)/(v(J) - 2) \leq 1$. On the other hand,

$$\frac{e(C_\ell) - 1}{v(C_\ell) - 2} = \frac{\ell - 1}{\ell - 2} > 1,$$

which implies $m_2(C_\ell) = (\ell - 1)/(\ell - 2)$. Now, let us analyze subgraphs of K_r. For each $J \subseteq K_r$, we have $e(J) \leq \binom{v(J)}{2}$. Thus,

$$\frac{e(J) - 1}{v(J) - 2} \leq \frac{\binom{v(J)}{2} - 1}{v(J) - 2} = \frac{v(J) + 1}{2},$$

for each $J \subseteq K_r$ such that $v(J) \geq 3$. It follows that $m_2(K_r) = (r + 1)/2$. Next, for each $\ell \geq 3$, consider the function $f_\ell : \mathbb{N} \to \mathbb{Q}$ defined by

$$f_\ell(t) = \frac{\binom{t}{2}}{t - 2 + m_2(C_\ell)^{-1}}.$$

It is not hard to check that $(f_\ell(t))_{t\geq 3}$ is monotone increasing (for every given ℓ). Since $m_2(C_\ell) = (\ell-1)(\ell-2)$, we have

$$m_2(K_r, C_\ell) = f_\ell(r) = \frac{\binom{r}{2}}{r-2+(\ell-2)/(\ell-1)}. \tag{A1}$$

It follows readily from this identity that $m_2(K_r, C_\ell)$ is strictly decreasing in ℓ, and thus,

$$m_2(K_r, C_\ell) \leq m_2(K_r, C_3) = \frac{r(r-1)}{2r-3} < \frac{r+1}{2} = m_2(K_r), \tag{A2}$$

for every $r \geq 4$. Finally, the identity in (A1) implies that

$$m_2(K_r, C_\ell) = \frac{\binom{r}{2}(\ell-1)}{(r-1)(\ell-1)-1} = \frac{r}{2}\cdot\frac{1}{1-\frac{1}{(r-1)(\ell-1)}} > \frac{r}{2}.$$